

FILE I/O BEHAVIOR OF MODERN APPLICATIONS,

Tyler R. Harter, Michael B. Vaughn, Remzi H. Arpaci-Dusseau*,

University of Wisconsin-Madison, Department of Computer Science, Madison,
WI 53715, remzi@cs.wisc.edu

In today's world, personal computers are ubiquitous, and for many home users media-rich workloads are becoming extremely common. Despite this trend, much of the existing literature on file system performance focuses on classic business and server workloads while relatively little work is done on the file system utilization of home users. We analyze file system workloads generated by Apple Computer's popular iLife (iMovie, iPhoto, and iTunes) and iWork (Pages, Numbers, and Keynote) software suites. We find that accesses to large media files dominate the bytes that are read and written by iLife's multimedia applications. Contrary to expectations, the patterns of access to these media files are classified as random by naive definitions of sequentiality. We consider other definitions that allow us to better characterize such accesses. Another trend in modern computing is that applications often rely on a variety of high level APIs for file manipulation instead of using system calls to interact with the file system more directly. We encountered several instances where the abstractions of the intermediate API layer led to I/O behavior that was unlike the behavior one would expect if the applications' programmers had used the Unix file system calls directly.